TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

"Construcción de un invernadero con un sistema de control y monitorización de las variables del entorno"

Curso 2016/2017

Alumno/a:
Germán Ruano García

Director/es:
José del Sagrado Martínez

Resumen

El principal objetivo de este TFG es el de la creación de una maqueta de un invernadero que nos permita automatizar ciertas tareas, además de poder visualizar y controlar las variables del entorno.

Para su funcionalidad se ha desarrollado un sistema IOT el cual tiene como núcleo una Raspberry Pi actuando como servidor, y para su control y monitorización se ha desarrollado una aplicación para dispositivos Android.

En el proyecto se han explicado los distintos componentes, tanto “hardware” como “software” que han sido necesarios para la realización del mismo, además de los mapas de conexiones y diagramas de circuitos diseñados para este sistema.

Abstract

The main objective of this TFG is the creation of a model of a greenhouse that allows us to automate certain tasks, besides being able to visualize and control the variables of the environment.

For its functionality I have developed an IOT system which has as core a Raspberry Pi acting as a server, and for its control and monitoring I have developed an application for Android devices.

In this project are explained the various components, both "hardware" and "software" that have been necessary for its realization, as well as the connection maps and circuit diagrams designed for this system.

ÍNDICE

1. INTRODUCCIÓN.

La agricultura tradicional ha sido desde sus orígenes una tarea que requiere una gran dedicación para la obtención de una cosecha. Originalmente, las cosechas dependían principalmente de las condiciones climáticas (elemento el cual no puede ser controlado por el ser humano) y, además, éstas limitaban los distintos tipos de cultivo según las diferentes estaciones del año.

De esta forma, el ser humano no era capaz de poder controlar de una forma eficaz la calidad o la producción de los cultivos.

A lo largo de la historia de la agricultura, bien por ensayo y error o por investigaciones sobre la materia, se ha ido evolucionando en esta área, produciéndose grandes avances desde su origen, como por ejemplo, la capacidad de poder alterar genéticamente las semillas en un laboratorio.

Este trabajo se centra en uno de los inventos más significativos para la agricultura desde sus inicios. Éste se trata del *invernadero*, el cual posibilita la producción en masa de ciertos cultivos, permitiendo almacenar el calor en el interior, mientras que a su vez recibe la luz solar.

Los principios de estas estructuras, la de los invernaderos, comienzan alrededor de 1850 en la horticultura neerlandesa. Se descubrió que el cultivo de uvas en invernaderos incrementaba el rendimiento al tener un clima cálido con una temperatura constante, es decir, las plantas crecían más rápidamente con temperaturas cálidas sin grandes oscilaciones en la temperatura. En España, debido a las condiciones climáticas de la costa mediterránea, se desarrolló una proliferación del cultivo en invernaderos (a finales de la década de los 70), y las provincias de Almería, Murcia, Alicante y Granada fueron las principales áreas de proliferación. En la costa almeriense se notó un impacto mayor, donde casi toda su superficie de costa está cubierta por el llamado "mar de plástico". De hecho, se trata una de las pocas construcciones visibles desde el espacio, debido a las dimensiones del área que ocupa. Un ejemplo claro del paisaje de invernaderos se puede encontrar en el Campo de Dalías y en el Campo de Níjar, ambos en los municipios almerienses de El Ejido y Níjar, respectivamente. Este tipo de cultivo bajo plástico se basó casi al cien por ciento en invernaderos tipo "parral". Cabe destacar que la extensión total de invernaderos en Almería asciende a las 30.230 hectáreas, lo que hace que esta región sea el lugar del mundo con una mayor superficie de invernaderos.

Desde aquel momento hasta ahora, han sucedido numerosos avances en los cultivos de invernadero, tales como plásticos con una mayor capacidad térmica o nuevos sistemas de riego. Pero no ha sido hasta los últimos años, cuando los avances en la informática se cruzaron con la agricultura, dando lugar de esta forma a la existencia de invernaderos cada vez más "inteligentes".

Gracias a dichos avances tecnológicos [15], al abaratamiento de ciertos componentes y a que la agricultura en nuestra provincia es predominante, surge la idea de este proyecto, la creación de un invernadero orientado al uso doméstico, el cual nos permita obtener producciones agrícolas a pequeña escala, reduciendo el esfuerzo y el tiempo necesario para su cuidado. Además, la idea principal es la de ofrecer una alternativa al típico huerto "casero" que se estilaba antiguamente, ofreciendo la oportunidad de generar una pequeña producción de subsistencia.

1.1. Motivación personal.

A lo largo de mi estancia en el Grado de Ingeniería Informática, obtuve cierta formación en el ámbito del Internet de las Cosas (IOT) [4]. A raíz de dicho aprendizaje recibido, aumentó mi curiosidad por este campo de la informática, la cual me llevo a desarrollar mis conocimientos sobre el “IOT” de forma autodidacta.

El origen motivacional de la idea que da vida a este proyecto, surgió de las ganas de poner en práctica mis conocimientos adquiridos en el campo anteriormente comentado, unido al incentivo del gran sector de la agricultura en invernadero llevada a cabo en la provincia de Almería.

Debido a que esta idea se trataba de un proyecto personal, el cual pretendía realizar por “hobby”, y que además coincidía el momento de seleccionar un tema para el proyecto fin de grado, vi una gran oportunidad para efectuarlo en este trabajo.

Asimismo, este proyecto, me ha servido para adquirir nuevos conocimientos en el campo del “IOT”, los cuales considero que me serán de una gran utilidad en mi trayectoria profesional, ya que a raíz de comenzar este proyecto continuaré desarrollando mis conocimientos y formación en el campo del “IOT” en un futuro próximo.

1.2. Objetivos.

El objetivo de este proyecto es el de maximizar la producción y la calidad de ésta en un entorno que podamos controlar, facilitando así mismo la complejidad de las tareas y al mismo tiempo reduciendo el periodo necesario para realizarlas.

Esto es posible mediante un sistema automatizado [11] [12], el cual controla las variables naturales, como son la temperatura, la humedad y la luz, permitiendo adaptar estas variables al tipo de cultivo deseado.

Puntualizando los objetivos del proyecto, éstos se resumen en los siguientes puntos:

- Construcción de una maqueta donde se integre el sistema a desarrollar.
- Monitorizar la temperatura dentro de la instalación.
- Monitorizar la temperatura fuera de la instalación.
- Monitorizar la humedad del suelo.
- Monitorizar la humedad del ambiente dentro de la instalación.
- Monitorizar la humedad del ambiente fuera de la instalación.
- Automatización del sistema de riego.
- Automatización del sistema de iluminación.
- Automatización del sistema de ventilación.
- Controlar el sistema de forma remota desde un dispositivo “Android”.

1.3. Especificación de requisitos.

1.3.1. *Introducción.*

La funcionalidad de este proyecto, como se ha comentado anteriormente, se resume en poder monitorizar y controlar una instalación a pequeña escala de producción agrícola en invernadero desde un dispositivo “Android”.

1.3.1.1. Propósito.

El propósito de este proyecto se basa en ofrecer la capacidad de producir una limitada producción agrícola en una instalación de agricultura de forma que reduzca el esfuerzo necesario para su cuidado.

1.3.1.2. Ámbito del sistema.

Este sistema será operativo remotamente desde cualquier lugar del mundo en el que tengamos acceso a Internet, con la condición de que la infraestructura también posea una conexión red. De esta forma, la instalación y el dispositivo estarán siempre conectados intercambiando la información necesaria para su correcto funcionamiento.

Cabe añadir que el sistema está enfocado para un entorno de producción agrícola a pequeña escala, de forma que está orientado para un uso "casero".

1.3.1.3. Definiciones, Acrónimos y Abreviaturas.

- *Arduino* [1] – Es una plataforma de electrónica "open-source" o de código abierto cuyos principios son contar con software y hardware fáciles de usar.
- *IOT* – Es un concepto que se refiere a una red de objetos cotidianos conectados a través de Internet, lo que les permitirá recolectar e intercambiar datos de forma constante. Esta tendencia ha sido llamada "La siguiente Revolución Industrial", debido al impacto que se espera que tenga esta sobre la forma en la que la gente vive, trabaja, viaja, se entretiene e interactúa con las otras personas, los negocios y los gobiernos alrededor del mundo.
- *Higrómetro* [3] – También llamado higrógrafo, se trata de un instrumento utilizado para medir el grado de humedad del aire o de otros gases. En la meteorología es un instrumento que se usa para medir el contenido de humedad en la atmósfera.
- *MySQL* [5] - Se trata de una aplicación con la cual se pueden gestionar archivos llamados desde bases de datos.
- *Script* [7] – Consiste en un documento, el cual contiene instrucciones, que están escritas en códigos de programación. Es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa de un ordenador.
- *Servidor Web* [8] – Se trata de un programa informático el cual procesa una aplicación desde el lado del servidor, y a su vez realiza conexiones bidireccionales o unidireccionales con el cliente, y genera una respuesta en cualquier lenguaje.
- *Raspberry* – Se trata de un computador de placa reducida de bajo costo que soporta varios componentes necesarios en un ordenador común. Tal y como dicen

en la página web del producto *"Es un pequeño ordenador capaz, que puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición"* (Rapsberry Pi Foundation).

1.3.2. *Descripción general.*

1.3.2.1. Funciones del sistema.

El sistema debe ser capaz de monitorizar las variables de humedad, temperatura y nivel del depósito de agua, tanto como controlar los sistemas de iluminación, riego y ventilación, todo ello a través de un dispositivo "Android". Además, permitirá la visualización del interior de la estructura en directo mediante una cámara.

1.3.2.2. Restricciones.

Para la elaboración de este proyecto se ha tenido que aplicar el desarrollo con dos restricciones básicas, las cuales han obligado a simplificar la idea original del proyecto. Dichas restricciones son:

- *Económicas:* Dado que este proyecto ha sido financiado por mí en su totalidad, no se han podido introducir tantos sistemas de control del entorno como se hubiese deseado, a la vez que ciertos componentes no tienen la precisión óptima para enfocarlo a una puesta en explotación en un ámbito industrial.
- *Tiempo:* Al tener fechas límites para la elaboración del proyecto, no se ha podido extender tanto como se hubiese deseado, teniendo que adaptar el proyecto al tiempo disponible, por lo que se ha reducido notablemente su envergadura original.

1.3.2.3. Suposiciones y dependencias.

El diseño del sistema está enfocado al uso de un solo dispositivo por instalación, de forma que un único usuario controle el sistema con su teléfono personal.
Si se utilizase en más de un dispositivo el sistema de notificaciones, por ejemplo el de nivel de agua en el depósito de agua, no funcionaría correctamente.

1.3.3. Requisitos específicos.

1.3.3.1. Requisitos funcionales.

A continuación se detalla la lista de requisitos funcionales del sistema, identificando cada requisito con un título y detallando debajo de éste la descripción del mismo:

- *Monitor de temperatura.*
  El sistema estará dotado de una funcionalidad que permita la medición en tiempo real de la temperatura, tanto del interior de la estructura como fuera de ella.
- *Monitor de humedad ambiental.*
  El sistema estará dotado de una funcionalidad que permita la medición en tiempo real de la humedad, tanto del interior de la estructura como fuera de ella.
- *Monitor de humedad del suelo.*
  El sistema estará dotado de una funcionalidad que permita la medición en tiempo real de la humedad del suelo.
- *Monitor de nivel de depósito.*
  El sistema estará dotado de una funcionalidad que permita la medición en tiempo real de la reserva de agua disponible en el depósito.
- *Monitor de sistemas.*
  El sistema estará dotado de una funcionalidad que permita la visualización de los estados de los sistemas de riego, luz y ventilación.
- *Notificaciones de niveles críticos en depósito.*
  El sistema deberá notificar al usuario, mediante notificaciones "push", cuando los niveles de agua del depósito bajen a niveles críticos, los cuales serán considerados de esta forma con un aforo del 20% de la capacidad y posteriormente el 10% de la misma.
- *Sistema de riego.*
  El sistema deberá incluir una funcionalidad que permita activar el riego en dos modos distintos:
  - Automático: Se establece una humedad objetivo, la cual se mantendrá siempre en ese nivel, es decir, regará hasta alcanzar el nivel y una vez ese nivel descienda, el sistema volverá a activarse.
  - Programado: Se establece una hora y un tiempo de duración, de forma que cada día, a la hora indicada, se activará el sistema durante el tiempo indicado.

- *Sistema de iluminación.*

  El sistema deberá incluir una funcionalidad que permita activar la iluminación en dos modos distintos:

  - Manual: El sistema permanecerá encendido o apagado, según lo indique el usuario en la aplicación.
  - Programado: Se establece una hora y un tiempo de duración, de forma que cada día, a la hora indicada, se activará el sistema durante el tiempo indicado.

- *Sistema de ventilación.*

  El sistema deberá incluir una funcionalidad que permita activar la ventilación en dos modos distintos:

  - Manual: El sistema permanecerá encendido o apagado, según lo indique el usuario en la aplicación.
  - Programado: Se establece una hora y un tiempo de duración, de forma que cada día, a la hora indicada, se activará el sistema durante el tiempo indicado.

- *Sistema de video vigilancia.*

  El sistema deberá incluir una funcionalidad que permita visualizar lo que ocurre dentro de la estructura en tiempo real.

- *Accesibilidad remota.*

  El sistema deberá permitir el funcionamiento correcto desde cualquier lugar donde se disponga de conexión a Internet.

1.3.3.2. Requisitos no funcionales.

A continuación, se detalla la lista de requisitos no funcionales del sistema, organizados por categorías, detallando debajo del mismo su descripción:

- *Eficiencia.*

  El sistema debe ser capaz de actualizar los valores en la base de datos al menos una vez cada 30 segundos.

- *Usabilidad.*

  El sistema deberá incluir mensajes de aviso en las interacciones con el usuario, de forma que si por ejemplo, activa un sistema, reciba un mensaje de que ha sido

activado. Además, el sistema deberá incluir iconos o imágenes en la aplicación, de forma que ayude a identificar los elementos de control y monitoreo.

- *Disponibilidad.*

  El sistema deberá ser capaz de estar operativo 24 horas al día los 7 días de la semana.

- *Plataforma.*

  La aplicación del sistema será desarrollado para dispositivos móviles “Android”.

1.4. Planificación temporal.

Para la realización de este proyecto se ha elaborado una planificación temporal de carácter aproximado. Esta planificación se plasma en un diagrama de Gantt.

En primer lugar, tenemos una tabla (Tabla 1.4.1.) con las tareas a realizar y el número de días estimado con un trabajo estimado de 8 horas diarias:

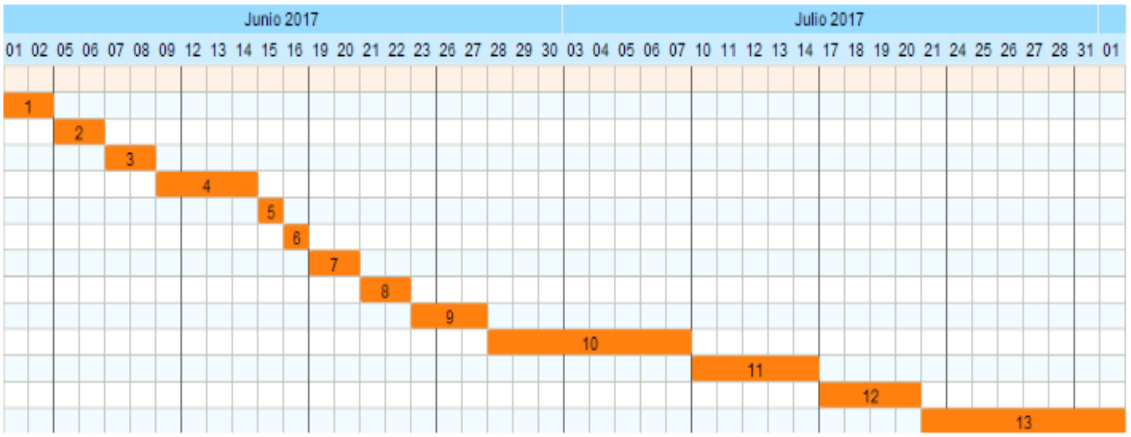
Tabla 1.4.1. “Tareas a realizar y número de días estimado”

Id	Tarea	Días
1	Formulación del problema	2
2	Estudio de tecnologías “controladores”	2
3	Estudio agricultura de invernadero	2
4	Diseño y construcción de maqueta	4
5	Diseño e implementación de base de datos	1
6	Diseño circuito electrónico	1
7	Codificación de controlador	2
8	Implementación de circuito	2
9	Diseño de APP	3
10	Codificación de APP	8
11	Pruebas y correcciones	5
12	Evaluación de resultados	4
13	Elaboración de memoria	8

Fuente: Elaboración propia.

En segundo lugar, se muestra un calendario (Gráfico 1.4.2.) dónde se representan las tareas con su duración estimada, pudiendo identificar dichas tareas mediante el identificador escrito en la tabla anterior, el cual es el que corresponde con su lugar en el calendario:

Gráfico 1.4.2. "Calendario de las tareas"



Fuente: Elaboración propia.

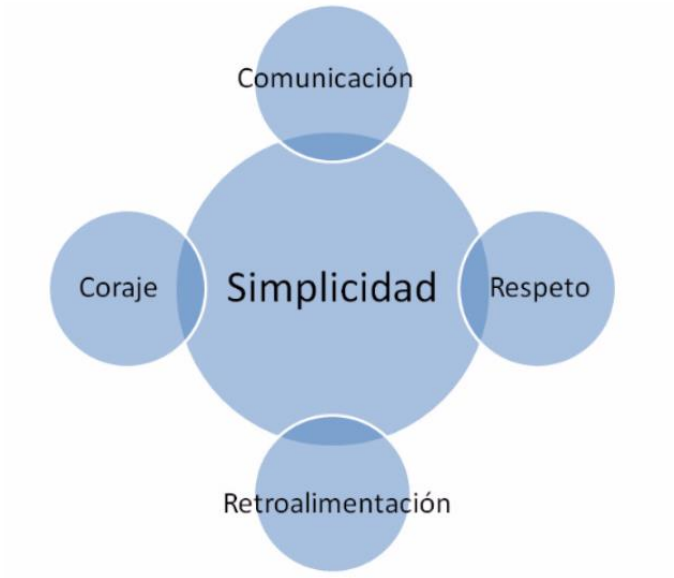
2. MÉTODOLOGÍA.

Para este proyecto, se utilizará la metodología de desarrollo "Extreme programming (XP)", ya que al no tener experiencia en este tipo de proyectos, se requerirá un mayor enfoque en la adaptabilidad, produciéndose cambios sobre la marcha.

Esta metodología se caracteriza por tener un desarrollo iterativo e incrementado, en el cual se realizan continuas pruebas unitarias durante todo el proceso.

Su mayor *ventaja* es la capacidad de adaptación tanto a grandes como a pequeños proyectos. Y encontramos entre sus *desventajas* que no se tiene una definición del coste y del tiempo de desarrollo.

Encontrándose entre sus cinco valores los siguientes:



- *Comunicación.*

  Todos son parte del equipo y se comunican cara a cara todos los días. Se trabaja juntos en todo, desde los requerimientos hasta la programación. En equipo se creará la mejor solución al problema.

- *Simplicidad.*

  Se desarrolla lo que se ha solicitado y lo necesario, pero no más de eso. De esa forma, se maximiza el valor de la inversión realizada. Se dirige al objetivo a pasos simples y pequeños, mitigando los fallos a medida que ocurran. Se creará algo que pueda mantenerse en el largo plazo y a costos razonables.

- *Retroalimentación.*

  Se tomará seriamente los compromisos con el usuario establecidos en todas las iteraciones, entregando software en funcionamiento en cada una. Se mostrará al usuario el software frecuentemente y de forma temprana, escuchando cuidadosamente sus observaciones y realizando los cambios que sean necesarios. Se adaptarán los procesos al proyecto y no al contrario.

- *Coraje.*

  Se dirá la verdad en los avances y estimados, no se documentarán excusas para el fracaso, pues se planificará para tener éxito. No se tendrá miedo a nada pues nadie trabaja solo. Se adaptará a los cambios cuando sea que estos ocurran.

- *Respeto.*

  Todos en el equipo dan y reciben el respeto que merecen como integrantes del equipo, y los aportes de cada integrante son valorados por todos. Todos contribuyen, así sea simplemente con entusiasmo. Los desarrolladores respetan la experticia de los clientes y viceversa. La Gerencia respeta el derecho del equipo de asumir responsabilidad y tener autoridad sobre su trabajo.

3. MATERIALES.

En este capítulo se pretende recopilar una lista de todos los componentes utilizados para la elaboración de este proyecto. Esta lista incluirá, tanto componentes "hardware" como "software". Además, se documentará el presupuesto estimado para la realización de dicho proyecto.

3.1. Componentes "hadware".

3.1.1. Raspberry pi 2B.

¿Qué es?

La descripción genérica de este artilugio se ha descrito en el "punto 1.4.1.3", pero para este proyecto he usado la Raspberry pi 2B [6], la cual se trata de la segunda generación de este producto, y por lo tanto contiene ciertas mejoras que su pionera, y estas mejoras son: 1 GB de RAM (en vez de 512 MB) y una procesadora quad-core ARM Cortex-A7 de 900MHz (en vez de ARMv6 de 700 MHz).

En este proyecto, actuará como servidor de todo el proyecto. También será el dispositivo encargado de activar mediante sus pines los sistemas de ventilación, de riego y de iluminación.

¿Por qué lo utilizo?

Bien se podría usar cualquier otro mini-ordenador que se pudiese integrar físicamente en el proyecto, pero ya poseía este dispositivo, por lo que ha significado un ahorro en los costes del proyecto.

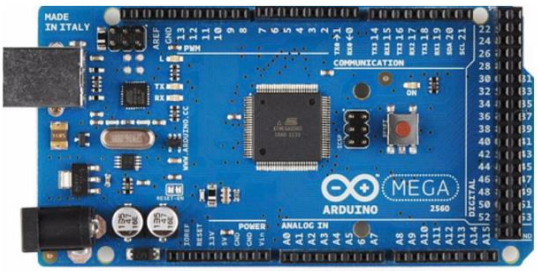
3.1.2. Controlador ATmega2560.

¿Qué es?

Se trata de un micro controlador. En este proyecto se pretende utilizarlo para la recolección de datos de sensores, tanto analógicos como digitales.

¿Por qué se utiliza?

La justificación del uso de este controlador se basa en que se requerían de 8 pines analógicos para la lectura de ciertos datos, descartando de esta forma el uso directo del dispositivo Raspberry. Además, este micro controlador es uno de los pocos que poseen al menos 8 pines analógicos.



3.1.3. Módulo ENC28j60.

¿Qué es?

ENC28J60 es un controlador de Ethernet, diseñado para sistemas embebidos, fabricado por Microchip Technology Inc. Este puede ser utilizado en conjunto con micro controladores como el ATmega2560, permitiendo una comunicación Ethernet con otros

dispositivos. En el caso de este proyecto se comunica con un servidor, enviando a éste las mediciones realizadas por los sensores.

¿Por qué lo utilizo?

ATmega2560 no dispone de ningún tipo de conexión Ethernet que permita enviar los datos medidos al servidor, por lo que se necesita un módulo adicional que nos permita este tipo de conexión. Inicialmente se iba a utilizar el módulo de red inalámbrica ESP8266, ya que nos permitiría mantenernos conectados mediante Wifi. Finalmente, se decidió usar ENC28j60, ya que al tratarse de componentes tan económicos, se requería de una conexión estable que estuviese exenta de posibles micro-cortes o pérdida de visibilidad del punto de acceso.

3.1.4. Relé.

¿Qué es?

Se trata de un dispositivo rudimentario que permite alternar los caminos que seguirá una corriente eléctrica en un circuito. Consiste en un pequeño electroimán que al mandarle un estímulo eléctrico se activará, moviendo una "palanca" que hará contacto con otro borne, haciendo así que la corriente circule por otro camino. Esto nos permite manejar corrientes eléctricas de gran voltaje mediante, por ejemplo, un micro controlador como el nuestro, que tan solo puede manejar internamente tensiones de 5V.

¿Por qué lo utilizo?

Como ATmega2560 solo puede manejar tensiones de 3,3V o 5V, se requiere de este dispositivo para el manejo de las corrientes que circularán por los sistemas de ventilación, riego e iluminación, las cuales requieren tensiones de 12V, 12V y 220V respectivamente.

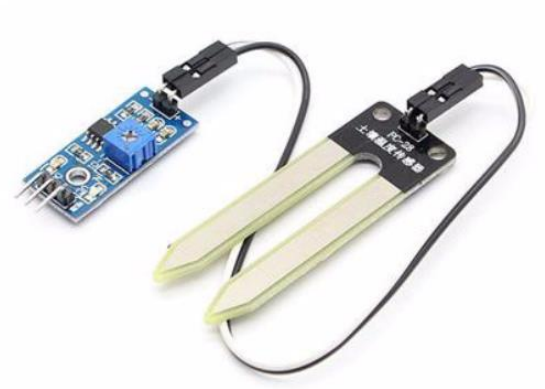
3.1.5. Higrómetro FC28.

¿Qué es?

Se trata de un sensor de humedad para el suelo. Éste envía una señal analógica, la cual variará en función de la cantidad de humedad a la que el sensor esté expuesto.

¿Por qué lo utilizo?

Para la realización de este proyecto es necesario medir la humedad del suelo. Existen alternativas de mayor calidad, pero al verse limitado el proyecto económicamente, se ha optado por el uso de estos, los cuales son bastante económicos.



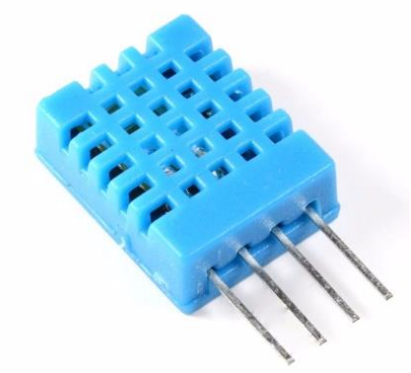
3.1.6. Sensor DHT11.

¿Qué es?

El DHT11 es un sensor digital capaz de medir tanto temperatura como humedad en el ambiente.

¿Por qué lo utilizo?

Existen sensores de este tipo con frecuencias de medición más elevadas y mayor precisión, pero ya se poseía este sensor, por lo que se ha decidido usar DHT11 para abaratar costes.



3.1.7. Sensor HC-SR04.

¿Qué es?

HC-SR04 es un sensor de distancia basado en ultrasonidos. Envía una onda de ultrasonidos, ésta rebota en el objeto más cercano, volviendo al sensor. De esta forma, en función del tiempo que transcurre desde que se envía hasta que recibe la onda, se puede calcular la distancia a la que reboto dicha onda.

¿Por qué lo utilizo?

El proyecto tiene un depósito de agua, el cual, en un determinado momento, termina por agotarse. Por ello, se requería de algún sistema que permitiese medir el nivel de agua restante en el depósito. De esta forma, se coloca este sensor en el techo del depósito a una distancia segura, permitiendo medir la distancia con el agua. Esto nos permite calcular de forma sencilla el % de agua que hay en el depósito.

Además de lo ya mencionado, se ha utilizado este sensor porque ya se poseía una unidad del mismo, lo cual ha significado un ahorro en costes de materiales de algún otro concepto de sistema de medición.

3.1.8. Logitech C270.

¿Qué es?

Cámara web USB marca Logitech.

¿Por qué lo utilizo?

El proyecto debe constar de un sistema de video vigilancia. Debido a que el sistema consta de pocas interfaces Ethernet, se ha descartado el uso de cámaras IP, utilizando en su lugar una cámara con interfaz USB. La justificación del uso de este modelo se basa en que Raspberry no es capaz de reconocer cualquier cámara USB.

Se poseía este modelo antes del inicio del proyecto y se comprobó que era compatible con Raspberry.

3.1.9. Ventilador 92mm.

¿Qué es?

Ventilador de 92mm de diámetro, alimentado a 5v o 12v.

¿Por qué lo utilizo?

Este proyecto debe incluir un sistema de ventilación. Se ha escogido el tamaño del ventilador estimando que era el necesario para el tamaño de la maqueta.

3.1.10. Fuente de alimentación ATX.

¿Qué es?

Fuente de alimentación ATX de PC.

¿Por qué lo utilizo?

Se requiere de una fuente de alimentación que suministre energía a todo el sistema, siendo esta la forma más económica de poder suministrar dicha energía.

3.1.11. Luces espectro completo 3W.

¿Qué es?

Se trata de una bombilla led que emite el espectro completo necesario para el crecimiento de las plantas.

¿Por qué lo utilizo?

El uso de esta bombilla en el proyecto se justifica básicamente para hacer una demostración de que el proyecto incluye un sistema de iluminación que funciona.

A usos prácticos, este tipo de luces son útiles para el cultivo en lugares donde el cultivo no reciba la luz suficiente o cultivos de interior.

Como en el caso de esta instalación se ha situado en una localización exterior con grandes cantidades de luz solar, se han utilizado en modo de ejemplo estas bombillas de tan poca potencia en modo de ejemplo, ya que 3W no son suficientes para suplir las necesidades de luz de un cultivo.

3.1.12. DC30E.

¿Qué es?

Se trata de una pequeña bomba de agua de 4,2W enfocada al uso de acuarios.

¿Por qué lo utilizo?

El proyecto incluye un sistema de riego, el cual debe ser propulsado por una bomba de agua. De esta forma, esta bomba fue el modelo seleccionado, ya que era el material más económico que suplía las necesidades para las dimensiones de la maqueta de este proyecto.

A continuación se muestra una tabla con el *presupuesto estimado* de estos componentes:

Nombre	Cantidad	Precio
Raspberry Pi 2b	1	35,04€
ATmega2560	1	6,92€
ENC28j60	1	2,35€
Modulo Relé	3	2,76€
Higrómetro FC-28	8	8,22€
Sensor DHT11	3	2,10€
Sensor HC-SR04	1	1,19€
Logitech C270	1	26,80€
Ventilador 92mm	2	13,62€
Fuente alimentación	1	15,95€
Bombillas 3w	3	5,28€
DC30E	2	7,16€
Materiales de estructura	-	≈50,00€
Total	-	**≈177,39€**

3.2. Componentes “software”.

3.2.1. Android Studio.

¿Qué es?

Se trata de un entorno de desarrollo integrado, el cual se basa en IntelliJ IDEA de la compañía JetBrains. Enfocado en el desarrollo de aplicaciones en “Android”.

¿Por qué lo utilizo?

Ya que el proyecto requiere de la programación de una aplicación en “Android”, se ha escogido este programa. Se ha decidido utilizar este entorno, ya que el plugin ADT para eclipse ha dejado de tener soporte, convirtiendo Android Studio en la opción más viable.

3.2.2. PHPMyAdmin.

¿Qué es?

PhpMyAdmin es una herramienta que nos permite administrar bases de datos MySQL directamente desde un navegador web.

¿Por qué lo utilizo?

Se ha decidido utilizar herramienta ya que el servidor es una Raspberry, y PhpMyAdmin es la herramienta de este tipo más utilizada en este entorno y por lo tanto, hay mucha documentación accesible.

3.2.3. MySQL.

¿Qué es?

Es un sistema de administración de bases de datos para bases de datos relacionales.

¿Por qué lo utilizo?

Ante la necesidad de crear una base de datos para el proyecto, se decidió utilizar MySQL porque es un entorno con el que ya estaba bastante familiarizado.

3.2.4. MJPG-Streamer.

¿Qué es?

MJPG-Streamer es una aplicación de líneas de comandos, la cual permite capturar video desde un dispositivo, para mostrarlo a más de un dispositivo.

¿Por qué lo utilizo?

Se requiere de una herramienta que permita streaming del interior de la instalación y se ha escogido MJPG-Streamer, ya que es una herramienta bastante sencilla con mucha documentación en la red. Tampoco era necesario que el streaming reprodujese el audio del interior, por lo que esta herramienta cumplía con las funciones requeridas, siendo bastante sencilla de manejar [16].

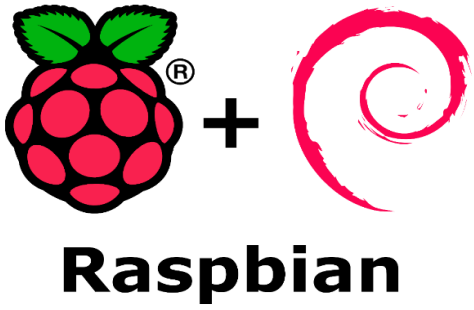
3.2.5. Rapsbian.

¿Qué es?

Se trata de una distribución Linux basado en Debian Jessie. Esta distribución ofrece un entorno de escritorio Linux originalmente en placas Raspberry y actualmente se está llevando a otros ordenadores embebidos con arquitecturas ARM.

¿Por qué lo utilizo?

Se utiliza este sistema operativo, ya que es el sistema operativo oficial de las placas Raspberry. De esta forma se asegura un correcto funcionamiento de la placa, sin comportamientos indebidos.



3.2.6. Arduino-IDE.

¿Qué es?

Arduino IDE es un entorno de desarrollo integrado utilizado para programar el comportamiento que deseemos que realice una placa Arduino o incluso clones de estas placas.

¿Por qué lo utilizo?

Se requiere de un entorno de desarrollo que permita la programación de la placa ATMega2560 utilizada en el proyecto, siendo Arduino IDE compatible con esta.

3.2.7. Apache Web Server.

¿Qué es?

Apache Web Server es un servidor web HTTP open source, para plataformas UNIX, Windows y Macintosh, el cual implementa el protocolo HTTP/1.1 y la noción de sitio virtual [14].

¿Por qué lo utilizo?

Existen distintas alternativas de servidores web HTTP para raspbian como Lighttpd o Nginx, pero como ya se había trabajado anteriormente con este servidor, se ha optado por usar esta opción, además de tener una gran cantidad de información disponible en la red.

3.2.8. Xampp.

¿Qué es?

XAMPP es un paquete de instalación de software libre, que contiene MySQL, Apache Web Server e intérpretes de lenguajes script PHP y Perl.

¿Por qué lo utilizo?

Este paquete incluye una lista de herramientas que se han mencionado anteriormente, pero en este caso se ha utilizado XAMPP en una PC Windows, con el objetivo de desarrollar los script PHP en un entorno de pruebas.

3.2.9. VCNViewer.

¿Qué es?

Se trata de un software libre, el cual está basado en una estructura cliente-servidor que nos ofrece la posibilidad de controlar un ordenador servidor a través de un ordenador cliente.

¿Por qué lo utilizo?

Las últimas versiones de Raspbian traen preinstalada esta herramienta. Las alternativas a VNCViewer ofrecían las mismas características, por lo que se ha optado por utilizar este programa. Se ha utilizado para mantener una conexión remota con el servidor Raspberry, con el objetivo de realizar todo el desarrollo del servidor.

3.2.10. No-IP.

¿Qué es?

No-IP es un servicio de DNS dinámico, el cual permite identificar una dirección IP pública con un nombre de dominio más sencillo de recordar, independientemente de tener o no IP estática [13].

¿Por qué lo utilizo?

Existen una gran cantidad de alternativas a este servicio, pero No-IP permite el uso de sus servicios de forma gratuita, obteniendo así un nombre de dominio. Su uso es necesario para permitir la conexión del sistema desde fuera de la red local en la que esté situado el invernadero.

3.2.11. EtherCard.

¿Qué es?

EtherCard es un driver de código abierto para el módulo de red ENC28j60. Este driver se puede encontrar en GitHub [2] en el siguiente enlace: https://github.com/jcw/ethercard

¿Por qué lo utilizo?

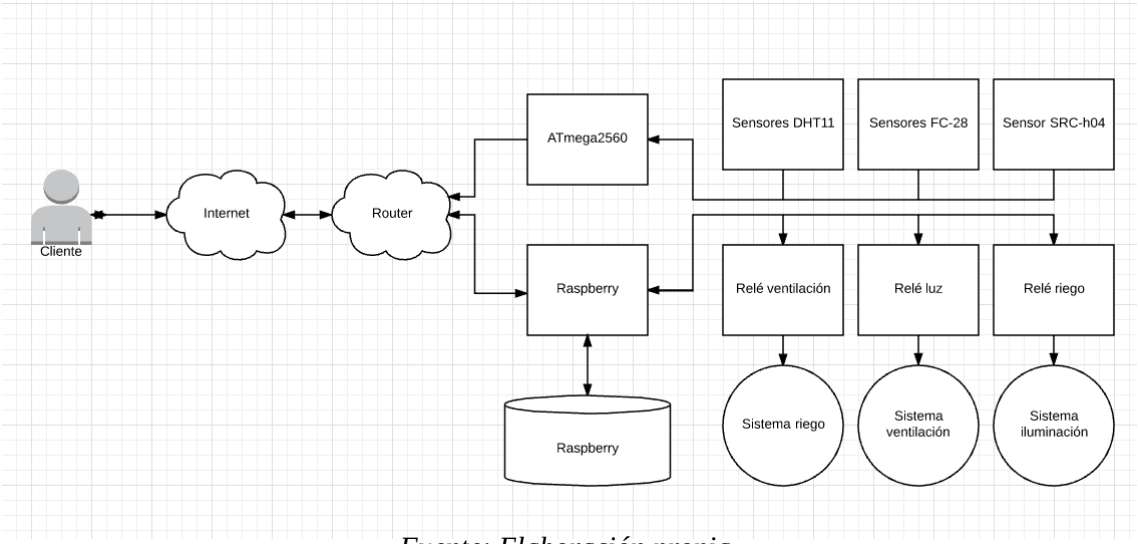
Se ha utilizado este driver para el módulo ENC28j60, ya que el driver incluido por defecto en Arduino IDE para este tipo de usos, no es compatible con este módulo. De esta forma, EtherCard ha sido la única opción que se ha encontrado para el manejo de dicho módulo.

4. INFRAESTRUCTURA.

En este capítulo se pretende recoger la información necesaria para entender el funcionamiento del proyecto que comprende la infraestructura y configuración, es decir, todo menos el sistema de control, el cual se verá más adelante.

Con el término “infraestructura” se hace referencia a la parte del proyecto que incluye la maqueta, servidor y controlador [10]. Excluyendo de esta parte la aplicación de control del sistema.

Para comenzar, la mejor forma de explicar el funcionamiento es con un esquema, el cual muestro a continuación:



Fuente: Elaboración propia.

En la imagen superior podemos visualizar el funcionamiento del sistema. El dispositivo raspberry actúa como servidor del sistema a la misma vez que activa o desactiva de forma directa los sistemas de riego, ventilación e iluminación. Toda la información que fluya en todo el sistema se almacenará o será procesada en este dispositivo. A la misma vez, se encarga de emitir un video en streaming y se encarga de enviar las notificaciones al cliente.

Por otro lado, está el dispositivo ATmega2560 unido al dispositivo ENC28j60, el cual le permite conexión Ethernet. Estos 2 dispositivos unidos se encargan de recoger las mediciones de los 3 tipos de sensores y enviarlos, mediante una llamada a un archivo PHP alojado en el dispositivo Raspberry. En esta llamada se le pasa una variable por método "GET", la cual contiene toda la información en formato de cadena de texto.

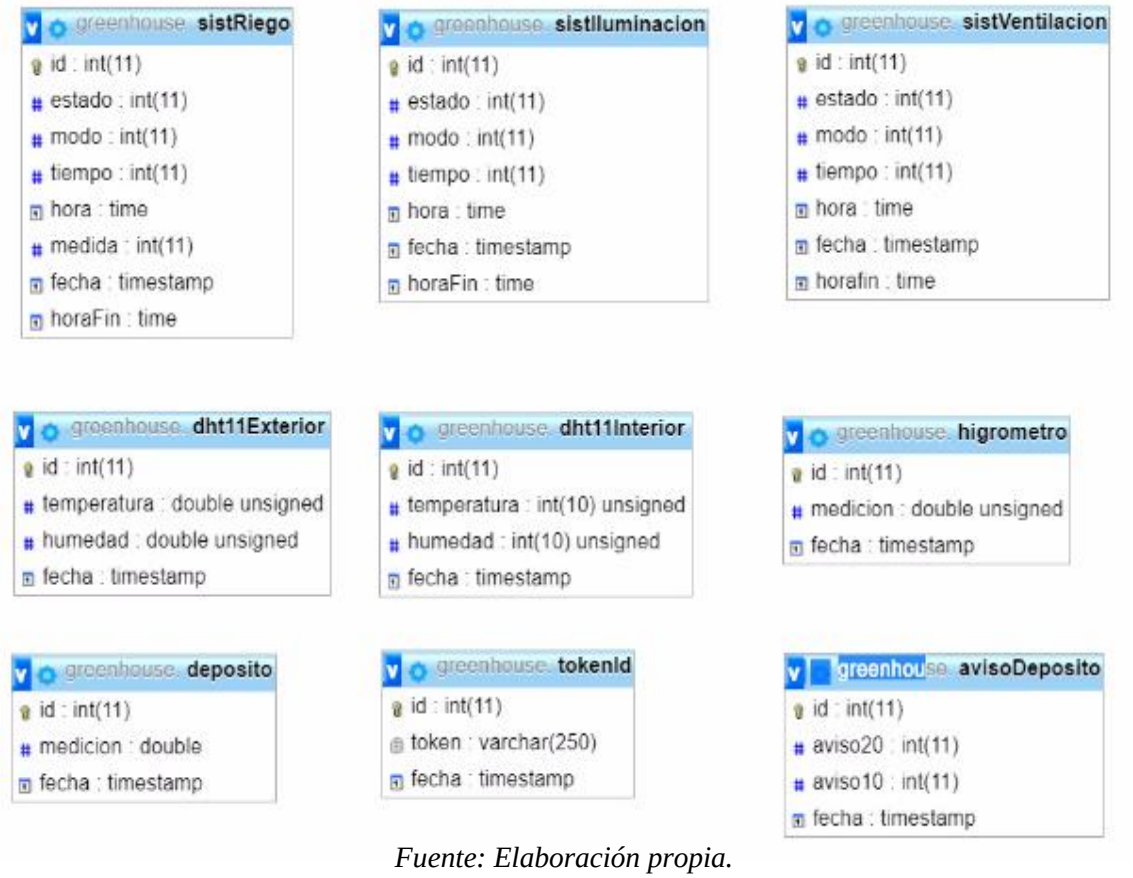
Una vez enviados datos desde ATmega2560 al archivo PHP del servidor, éste procesa la información y actualiza las tablas de la base de datos.

4.1. Base de datos.

Como se mencionó anteriormente, se ha utilizado MySQL para la base de datos del proyecto. Ésta, a su vez, ha sido gestionada mediante PHPMyAdmin.

A continuación, se muestra un esquema (Tabla 4.1.1.) de la base de datos, seguido de una explicación del uso de cada tabla y sus campos:

Tabla 4.1.1. "Esquema de la base de datos"



Fuente: Elaboración propia.

- **sistRiego**. Esta tabla se utiliza para el sistema de riego del proyecto. Con esta tabla podemos establecer cuál de los 2 modos de riego queremos seleccionar, en caso de ser modo *automático*, guardará el estado ideal de humedad que seleccionemos, y en caso de ser *programado*, guardará la hora a la que queremos que se active y la duración del riego.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
estado	Entero	Almacena el estado del sistema. Cuando vale 0 estará apagado, y cuando valga 1 estará encendido.
modo	Entero	Almacena el modo de funcionamiento. Cuando vale 1 estará en modo automático y cuando vale 2 estará en modo programado.
tiempo	Entero	Este campo guarda el tiempo en segundos que durará el riego en modo programado.
hora	Time	Almacena la hora a la que debe activarse el riego en modo programado.
medida	Entero	Guarda el valor óptimo indicado para el modo de riego.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

horaFin	Time	Guarda la hora a la que el sistema de riego debe apagarse después de activarse en modo programado.

- **sistIluminacion**. Esta tabla permite la activación del modo de iluminación en 2 modos: el modo *programado*, en el cual se indicará una hora del día y una duración; y el modo *manual*, el cual permanecerá apagado o encendido hasta que se indique lo contrario.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
estado	Entero	Almacena el estado del sistema. Cuando vale 0 estará apagado y cuando valga 1 estará encendido.
modo	Entero	Almacena el modo de funcionamiento. Cuando vale 1 estará en modo manual y cuando vale 2 estará en modo programado.
tiempo	Entero	Este campo guarda el tiempo en segundos que durará el sistema de iluminación en modo programado.
hora	Time	Almacena la hora a la que debe activarse el sistema de iluminación en modo programado.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila
horaFin	Time	Guarda la hora a la que el sistema de iluminación debe apagarse después de activarse en modo programado.

- **sistVentilacion.** En esta tabla se almacena la información necesaria para hacer funcionar el sistema de ventilación, el cual al igual que el sistema de iluminación, permite un modo programado y otro manual, los cuales funcionarán de la misma forma.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
estado	Entero	Almacena el estado del sistema. Cuando vale 0 estará apagado y cuando valga 1 estará encendido.
modo	Entero	Almacena el modo de funcionamiento. Cuando vale 1 estará en modo manual y cuando vale 2 estará en modo programado.
tiempo	Entero	Este campo guarda el tiempo en segundos que durará el sistema de ventilación en modo programado.
hora	Time	Almacena la hora a la que debe activarse el sistema de ventilación en modo programado.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.
horaFin	Time	Guarda la hora a la que el sistema de ventilación debe apagarse después de activarse en modo programado.

- **dht11Exterior**. Esta tabla se actualiza cada vez que el micro controlador hace una medición y la envía. La información que se guarda en esta tabla se trata de las mediciones del sensor dht11 situado en el exterior de la estructura. Se almacenan tanto la humedad como la temperatura.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
temperatura	Double	Guarda la temperatura medida por el sensor.
humedad	Entero	Guarda la humedad medida por el sensor.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

- **dht11Interior.** Exactamente igual que la tabla anterior "dht11Exterior", con la diferencia de que en este caso las mediciones provienen del interior de la estructura.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
temperatura	Double	Guarda la temperatura medida por el sensor.
humedad	Entero	Guarda la humedad medida por el sensor.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila

- **higrometro.** En esta tabla se almacenan las mediciones del sensor FC-28, el cual mide la humedad del suelo del interior de la estructura.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
medicion	Double	Guarda la humedad medida por el sensor.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

- **deposito.** Esta tabla es utilizada para almacenar las mediciones del sensor HC-SR04, el cual es utilizado para conocer el nivel de agua del depósito.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
medicion	Double	Guarda el nivel de agua restante.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

- **tokenId.** La tabla “tokenId” es utilizada exclusivamente para el envío de notificaciones. Sin esta tabla, si el id del dispositivo cambia, no podríamos enviar notificaciones al cliente, ya que el id al que estaríamos enviando ya no existiría.

Campo	Tipo	Descripción
Id	Entero	Identificador (primary key) de la fila en la tabla.
token	Varchar(250)	Guarda el id del dispositivo Android cliente.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

- **avisoDeposito.** Al igual que la tabla anterior “tokenId”, esta tabla también tiene un uso exclusivo para las notificaciones [9]. El proyecto incluye un sistema de notificaciones para el nivel de agua del depósito, las cuales se envían al descender del 20% y el 10%. Sin esta tabla sería imposible saber si dichas notificaciones han sido enviadas ya, por lo que se enviarán cada vez que midiese el nivel del depósito y estuviese por debajo del umbral. Con esta tabla, podemos saber si la notificación ha sido enviada ya, de forma que cuando se rellene el depósito y vuelva a superar los umbrales, volverá a activar las notificaciones que se enviarán cuando baje de nuevo el nivel.

Campo	**Tipo**	**Descripción**
Id	Entero	Identificador (primary key) de la fila en la tabla.
aviso20	Entero	Permite conocer si se ha enviado ya la notificación cuando el nivel del depósito baja del 20%. Cuando vale 0, la notificación no se ha enviado, y cuando vale 1, ya estará enviada.
aviso10	Entero	Permite conocer si se ha enviado ya la notificación cuando el nivel del depósito baja del 10%. Cuando vale 0, la notificación no se ha enviado, y cuando vale 1, ya estará enviada.
fecha	Timestamp	Guarda la fecha y hora a la que se añadió o fue modificada por última vez la fila.

4.2. Servidor Web.

El servidor web, el cual, como ya se ha mencionado anteriormente, se trata de “apache web server” instalado sobre un dispositivo “raspberry”.

Este servidor permite la conectividad de todo el proyecto, tanto del micro controlador que se encarga de subir los valores la base de datos, como de la aplicación del cliente, la cual

solicita estos datos y algunos más, a la vez que envía los modos de funcionamiento de los sistemas de riego, iluminación y ventilación.

Todas estas conexiones entre los dispositivos funcionan gracias a una serie de ficheros programados en PHP y HTML, los cuales a continuación se listan con una explicación de su uso:

- **config.php.** Este fichero es bastante sencillo. Es requerido por la mayoría de los demás ficheros, y su función consiste en abrir una conexión con la base de datos del servidor, por lo tanto contiene la información necesaria para establecer dicha conexión.
- **camara.html**. Este fichero HTML no tiene una función vital para el proyecto, sino que es una función programada para una visualización mejor de la cámara de seguridad. Hace una llamada al servicio de streaming de la cámara de seguridad, adaptado la visualización de esta a las dimensiones del dispositivo.
- **sistemas.php.** Aquí se indica el funcionamiento de los sistemas de riego, ventilación e iluminación. La aplicación cliente selecciona los parámetros de funcionamiento de éstos y los envía mediante una llamada a este archivo, el cual se encarga de guardar los valores que recoge en la base de datos.
- **controlador.php.** Este fichero complementa al fichero anterior "sistemas.php", el cual solo actualizaba los parámetros en la base de datos. Este fichero se encarga de leer estos parámetros que dictan el funcionamiento de los sistemas, una vez leídos se encarga de hacer funcionar los pines GPIO de la Raspberry que sean necesarios para el comportamiento que se ha indicado.
- **iot.php.** Este fichero es llamado desde el micro controlador ATmega2560. Cuando ATmega2560 llama a este fichero, proporciona una cadena de texto que contiene las mediciones realizadas con los sensores. Este fichero se encarga de convertir esa cadena de texto en valores numéricos para guardarlos en la base de datos.
- **valores.php.** Este fichero es llamado desde el cliente. Cuando es llamado, consulta en la base de datos los valores de los sensores para enviarlos al cliente, el cual se encargará de mostrarlos.
- **token.php**. Este fichero es llamado desde el cliente, el cual le proporciona el id del mismo dispositivo, además de ser necesario para la recepción de notificaciones. Una vez el archivo recibe el id, lo guarda en la base de datos.

- **notificacion10.php.** En este fichero se realiza el envío de una notificación al dispositivo cliente, enviándole un mensaje de que el depósito está al 10% de su capacidad.
- **notificacion20.php.** En este fichero se realiza el envío de una notificación al dispositivo cliente, enviándole un mensaje de que el depósito está al 20% de su capacidad.
- **notificacionDeposito.php**. Este fichero se encarga de leer de la base de datos el nivel de agua en el depósito. Cuando el nivel baja al 20% hace una llamada al fichero "notificacion20.php" y cuando baja hasta el 10% vuelve a hacer otra llamada, en este caso a "notificacion10.php". Una vez envía las notificaciones marca en la base de datos que han sido enviados, de forma que cuando los niveles vuelven a estar por encima del 10% y el 20% marca en la base de datos que se pueden volver a enviar las notificaciones cuando descienda de nuevo.

4.3. Scripts.

El correcto funcionamiento del servidor depende de que se ejecuten ciertos procesos en éste. Para que el sistema sea capaz de recuperarse de un corte de suministro eléctrico de forma autónoma, se han programado ciertos "scripts sh" que ponen en funcionamiento todos los procesos de forma autónoma al inicio del servidor.

Para ello se han dado permisos de ejecución a todos ellos. A continuación, se listan los archivos programados junto a una descripción de su funcionamiento en el proyecto:

- **streaming.sh.** Este script se encarga de ejecutar el servicio de MJPG-Streamer. Sin este script habría que ejecutar el servicio manualmente.
- **sistemas2.sh**. En este script se ejecuta de forma infinita cada 1 segundo. Hace una llamada al archivo "controlador.php", el cual como se mencionó anteriormente, se encarga de activar los sistemas de riego iluminación y ventilación en función de los parámetros introducidos. Haciendo de esta forma una llamada al archivo cada 1 segundo. Permitiendo así el funcionamiento de estos sistemas las 24h del día.
- **notificaciones.sh**. Al igual que el script "sistemas2.sh" su funcionamiento se basa en un bucle infinito. En este caso tenemos un bucle que descansa 5 segundos entre cada ejecución. En cada iteración hace una llamada al archivo

"notificacionDeposito.php", que como se mencionó anteriormente, es el fichero que se encarga de manejar las notificaciones del nivel del depósito de agua.

- **no-ip.sh.** Este script tiene como función la de ejecutar el servicio no-ip en el dispositivo. Sin este script habría que ejecutarlo manualmente en cada reinicio del servidor.

4.4. Medición de variables.

Como se ha descrito repetidamente a lo largo de esta memoria, una de las funcionalidades de este proyecto consiste en la medición de ciertas variables del entorno. Para la medición de estas variables, se han utilizado una serie sensores descritos anteriormente en el capítulo 3.

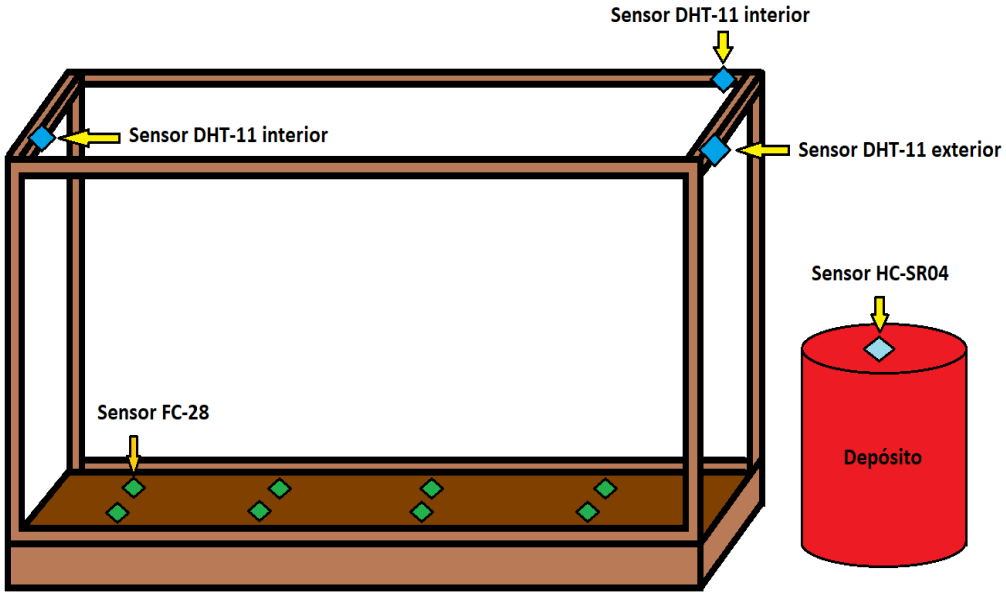
Las variables que se van a medir son las siguientes:

- Humedad interior.
- Humedad exterior.
- Humedad del suelo.
- Temperatura interior.
- Temperatura exterior.
- Nivel del depósito.

En la siguiente tabla se representa la relación de las variables y los sensores que las miden:

Sensor	Variable
DHT11	Humedad interior Humedad exterior Temperatura interior Temperatura exterior
FC-28	Humedad del suelo
HC-SR04	Nivel del deposito

Así mismo, se muestra a continuación una imagen con la disposición que mantienen en la maqueta los sensores nombrados en la tabla anterior:



Fuente: Elaboración propia.

Para la construcción de la maqueta, la cual tiene unas dimensiones de 60x85cm con una altura de 120cm, se ha diseñado la posición de los sensores de forma estratégica.

Como se ha comentado anteriormente, se han utilizado 3 sensores DHT-11, los cuales 2 han sido colocados en el interior de dicha maqueta de forma opuesta, con el objetivo de tomar mediciones más globales, y el tercer sensor ha sido colocado en el exterior.

Los sensores FC-28, los cuales son 8, han sido colocados en el suelo rodeando las plantas sembradas, de forma que les permita medir la humedad en los puntos más cercanos a las raíces.

Por último, el sensor HC-SR04 se ha colocado en lo alto del depósito orientado hacia el suelo, de forma que así pueda medir la distancia con el agua.

4.4.1. DHT11.

Para las mediciones de este sensor se ha utilizado la librería “DHT - sensor-library” de adafruit. Esta librería es de código abierto y puede encontrarse en GitHub en el siguiente enlace: https://github.com/adafruit/DHT-sensor-library.

Las variables medidas por este sensor se miden en las siguientes unidades:

Variable	Unidad
Temperatura	Grado Celsius (ºC)
Humedad	Humedad relativa (%)

A nivel de programación, en el código del micro controlador se debe incluir la librería incluyendo estos 2 archivos:

```
#include <DHT.h>
#include <DHT_U.h>
```

Para realizar la lectura de la humedad y temperatura basta simplemente con escribir el siguiente código:

```
DHT dht1(2, DHT11);
dht1.begin();
lectura=dht1.readHumidity();
humedad1=lectura;
lectura=dht1.readTemperature();
temperatura1=lectura;
```

Obteniendo así los valores en las unidades deseadas.

- Especificaciones técnicas.
    - Voltaje de funcionamiento 3 a 5V.
    - 2.5mA pico (durante solicitud de datos).
    - Bueno para lecturas de humedad entre 20-80% con un 5% de precisión.
    - Bueno para lecturas de temperatura entre 0-50°C con ±2°C de precisión.
    - Frecuencia de 1 Hz (una lectura por segundo).
    - Dimensiones 15.5mm x 12mm x 5.5mm.
    - 4 pines con espaciado de 0.1".

    El sensor dispone de 4 pines, de izquierda a derecha:

    - Pin 1: alimentación.
    - Pin 2: serial data.
    - Pin 3: vacío.
    - Pin 4: GND.

- Humedad y temperatura interior.

  Para la medición de esta variable se han utilizado 2 sensores del tipo DHT11. Estos 2 sensores están colocados de forma opuesta el uno del otro en el interior de la estructura. Haciendo esto, podemos obtener unas mediciones más próximas a los valores reales de las variables.

  Cuando el controlador ATmega2560 comienza a funcionar, realiza una medición con cada uno de los 2 sensores colocados en el interior. Una vez realizadas las mediciones con los 2, suma las 2 lecturas de humedad y por otra parte, las 2 lecturas de temperatura. Posteriormente estas 2 cifras se dividen entre 2, obteniendo así la media aritmética de las mediciones de los 2 sensores.

- Humedad y temperatura exterior.

  En esta medición tan solo se ha utilizado un sensor DHT11 ya que las variables del exterior no van a influir directamente en el desarrollo del hábitat interior, sino que se utilizan de una forma orientativa para comparar con los valores del interior de la estructura. De esta forma los valores mostrados serán directamente los que ofrece el sensor en su medición.

4.4.2. FC-28.

Para este sensor no se han utilizado librerías externas ya que se trata de un sensor analógico, es decir, mide señales analógicas que van desde 0 hasta 1023.

Este sensor carece de unidad de medida oficial, ya que la humedad se representa en un intervalo de 1-100 donde 1 equivale a una medición analógica del sensor de 1023 y donde 100 equivale a la medición analógica de 0.

- Especificaciones técnicas.
  - Voltaje de 3,3 a 5V cc.
  - Salidas analógica y comparadora.
  - Ajuste de sensibilidad.
  - Dimensiones del sensor 6x-20mm contactos 45mm.
  - Dimensión del comparador 30x14mm.

  El sensor dispone de 4 pines, de izquierda a derecha:
  - Pin 1: salida analógica.

- Pin 2: salida digital.
- Pin 3: alimentación
- Pin 4: GND.

- Humedad del suelo. Para la medición de esta variable se han utilizado 8 sensores repartidos por el suelo de la estructura.
  El micro controlador realiza las 8 mediciones de los respectivos sensores, una vez medidos, suma estos valores para posteriormente dividir la suma en 8, obteniendo la media aritmética de los valores.
  La fórmula para la representación del valor de la variable es la siguiente:

  100-(x*100/1024)

  Donde "x" representa el valor medido por el sensor.

4.4.3. HC-SR04.

Para el uso de este sensor se ha utilizado la librería "NewPing" que puede encontrarse en GitHub en el siguiente enlace: https://github.com/PaulStoffregen/NewPing.
Con esta librería se obtienen las mediciones en cm y en la representación de la variable se presenta con un porcentaje.
A nivel de programación, en el código del micro controlador se debe incluir la librería incluyendo el siguiente archivo:

```
#include <NewPing.h>
```

Para realizar la lectura de la distancia basta con escribir el siguiente código:

```
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
unsigned int uS = sonar.ping();
int distancia=sonar.convert_cm(uS);
```

- Especificaciones técnicas.
  - Voltaje 5V cc.
  - Consumo en reposo <2mA.
  - Consumo máximo 15mA.

- Angulo efectivo <15°.
- Rango de distancia 2cm – 400 cm.
- Resolución 0.3 cm.
- Angulo de medición 30º.
- Dimensión: 45mm x 20mm x 15mm.

El sensor dispone de 4 pines, de izquierda a derecha:

- Pin 1: alimentación.
- Pin 2: trigger (recoge la señal).
- Pin 3: echo(envía la señal).
- Pin 4: GND.

- Nivel del depósito. La medición de esta variable consiste en la colocación de este sensor en el techo del depósito, de forma que mida la distancia del tope del depósito con respecto al nivel del agua.

  La fórmula utilizada para obtener el nivel del depósito: de agua es la siguiente:

  x*100/(Dl-Dv)

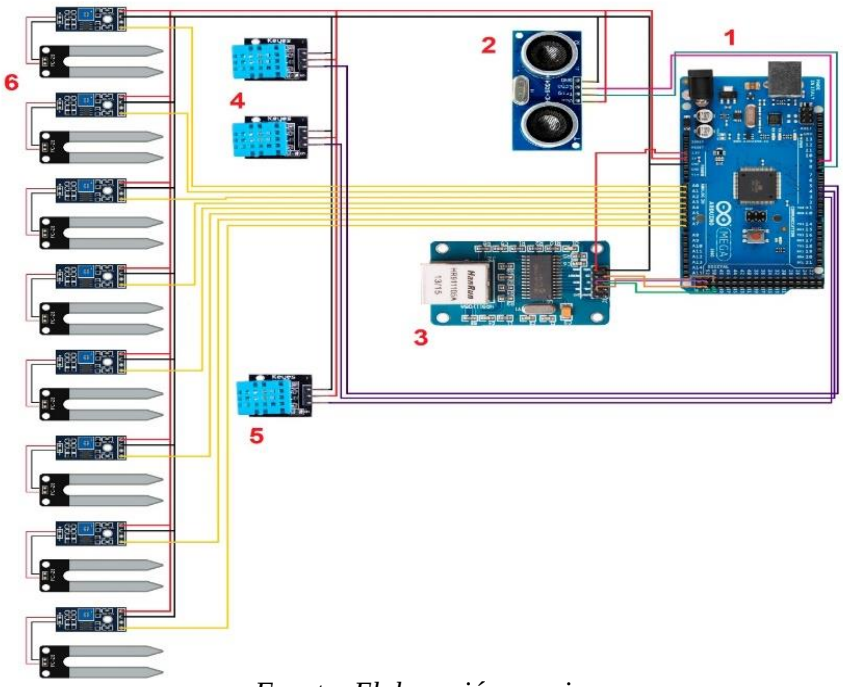
  Donde **x** = distancia medida por el sensor, **Dl** = distancia medida con depósito lleno y **Dv** = distancia medida con depósito vacío.

4.5. Mapa de conexiones.

En esta sección se pretende explicar gráficamente el diseño del circuito ideado para el proyecto, el cual se divide en dos secciones: la primera sección se trata del microcontrolador ATmega2560, y la segunda se centra en el mini pc Raspberry.

4.5.1. Conexiones ATmega2560.

En la siguiente imagen se puede observar gráficamente el mapeo de conexiones existente en el micro controlador:



Fuente: Elaboración propia.

En la tabla que se muestra a continuación (Tabla 4.5.1.1.) se indica el nombre de cada elemento correspondiente al número indicado en rojo de la imagen superior:

Tabla 4.5.1.1. "Nombre de los elementos existentes en el micro controlador".

Nº	Nombre
1	Micro controlador ATmega2560
2	Sensor HC-SR04
3	Módulo ENC28j60
4	Sensores DHT11 (Mediciones interior)
5	Sensor DHT11(Mediciones exterior)
6	Sensores FC-28

Fuente: Elaboración propia.

Para complementar la imagen, se muestra a continuación una tabla (Tabla 4.5.1.2.) con la correspondencia de las conexiones de los elementos con los pines utilizados en el micro controlador:

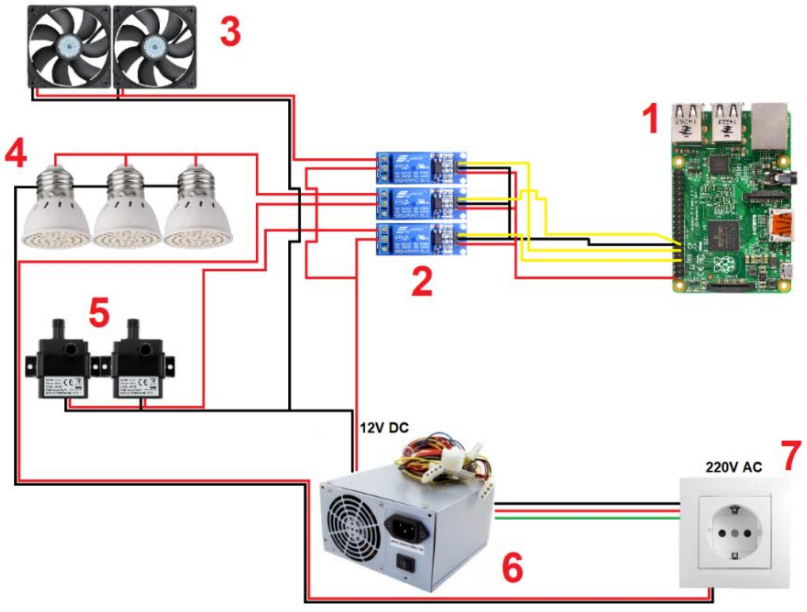
Tabla 4.5.1.2. "Correspondencia de las conexiones de los elementos con los pines usados en el micro controlador".

Elemento	Vcc	Pin	
ENC28j60	3,3V	SO	50
		SI	51
		SCK	52
		CS	53
HC-SR04	5v	TRIG	8
		ECHO	9
DHT11 (Interior)	5V	Nº1	4
		Nº2	5
DHT11 (Exterior)	5V	3	
FC-28	5V	Nº1	A0
		Nº2	A1
		Nº3	A2
		Nº4	A3
		Nº5	A4
		Nº6	A5
		Nº7	A6
		Nº8	A7

Fuente: Elaboración propia.

4.5.2. Conexiones Rapsberry.

Al igual que el micro controlador, la raspberry también posee una serie de conexiones, los cuales se representan en la siguiente imagen:



Fuente: Elaboración propia.

En la tabla que se muestra a continuación (Tabla 4.5.2.1.) se indica el nombre de cada elemento correspondiente al número indicado en rojo de la imagen superior:

Tabla 4.5.2.1. "Nombre de los elementos existentes en las conexiones de rapsberry".

Nº	Nombre
1	Raspberry pi 2b
2	Relés
3	Ventiladores
4	Bombillas espectro completo
5	Bombas de agua
6	Fuente de alimentación ATX
7	Enchufe corriente alterna

Fuente: Elaboración propia.

Para activar los relés que activarán los sistemas es necesario utilizar los pines GPIO de la raspberry, por ellos, en la siguiente tabla (Tabla 4.5.2.2.) se muestra el mapeo de conexiones:

Tabla 4.5.2.2. "Correspondencia de las conexiones entre los sistemas y los pines GPIO de la rapsberry".

Sistema	Vcc	Pin
Riego	5v	GPIO4
Ventilación	5v	GPIO17
Iluminación	5v	GPIO27

Fuente: Elaboración propia.

5. SISTEMA DE CONTROL.

Como se mencionó anteriormente, el proyecto está formado por la infraestructura, la cual ha sido detallada en el capítulo 4, y por el sistema de control, que en este proyecto se basa en una aplicación desarrollada para smartphones "Android".

Dicho sistema de control consta de 5 elementos:

- *Visualización de datos.*

  Este elemento tiene como única función visualizar el estado de todas las variables del sistema, pudiendo observar desde aquí los valores de las variables y el estado de los sistemas de riego, ventilación e iluminación.

- *Cámara de seguridad.*

  Este elemento también basa su funcionamiento en la visualización, pero en este caso se trata de la visualización de un vídeo en directo de lo que ocurre en el interior de la maqueta.

- *Sistema de ventilación.*

  Mediante este elemento del sistema de control se puede seleccionar como se debe comportar el sistema de ventilación. Se permiten 2 modos de funcionamiento: programado (se selecciona a qué hora debe encenderse y durante cuánto tiempo) y manual (encendido o apagado).

  Este sistema está formado por 2 ventiladores y un relé conectado a la Rasperry. La interacción con este elemento producirá que se encienda o apague el relé al que están conectados los ventiladores, haciéndolos funcionar o cesando su funcionamiento. Con esto se pretende tener un control sobre las variables de temperatura y humedad interior.

  La forma de interactuar con este sistema es la siguiente:

  Cuando se abre la ventana en la aplicación, en una lista desplegable, el usuario decidirá que modo desea utilizar, dependiendo cual seleccione se mostrara una pantalla u otra.

  - Modo manual:

    Se muestra un interruptor de 2 posiciones, identificado en color gris cuando se encuentra apagado y en verde cuando se encuentra conectado.

  - Modo programado:

    Se muestra una pantalla donde en primer lugar se podrá introducir la hora y minuto del día en que se pretende activar el sistema y en segundo lugar se deberán introducir los minutos y segundos que se desee que esté funcionando el sistema. Cuando se hayan seleccionado estos parámetros, se debe pulsar el botón "Guardar", el cual mandará la información a la base de datos.

- *Sistema de riego.*

  Parecido a los elementos de control de ventilación e iluminación, pero en este caso existen 2 modos de funcionamiento: automático (se establece un objetivo de humedad y se activa hasta que se alcanza) y programado (se le indica la hora a la

que debe activarse y la duración). Este sistema está compuesto por 2 bombas de agua conectadas a un relé.

Cuando este elemento se activa, incide sobre un relé, el cual hará funcionar o no las bombas de agua alojadas en el depósito de la infraestructura, haciendo que se riegue en el interior. Con la actuación de este sistema se ve afectada la variable de humedad del suelo.

La forma de interactuar con este sistema es la siguiente:

Cuando se abre la ventana en la aplicación, en una lista desplegable, el usuario decidirá que modo desea utilizar, dependiendo cual seleccione se mostrara una pantalla u otra.

- Modo automático.

  Se muestra una entrada en el que se debe seleccionar un número comprendido entre 1 y 100. Este número representa la humedad objetivo, cuando la humedad del suelo descienda por debajo del nivel indicado, se activará el relé, activando el sistema de riego, el cual no parara hasta que la humedad del suelo alcance el nivel indicado. Una vez seleccionado el objetivo de humedad se debe pulsar el botón "Guardar", el cual enviará la información a la base de datos.

- Modo programado.

  Se muestra una pantalla donde en primer lugar se podrá introducir la hora y minuto del día en que se pretende activar el sistema, y en segundo lugar se deberán introducir los minutos y segundos que se desee que esté funcionando el sistema. Cuando se hayan seleccionado estos parámetros, se debe pulsar el botón "Guardar", el cual mandará la información a la base de datos.

- *Sistema de iluminación.*

  Este elemento permite controlar el sistema de iluminación de la infraestructura, de una forma similar a la del sistema de ventilación, permitiendo los mismos modos de funcionamiento. Este sistema se compone por 3 bombillas y un relé.

  En este caso el relé actuará sobre las luces de la infraestructura. Con este sistema no se ve afectada ninguna variable, lo que se pretende es dotar de un espectro de luz completo a las plantas para incentivar su crecimiento.

  La forma de interactuar con este sistema es la siguiente:

Cuando se abre la ventana en la aplicación, en una lista desplegable, el usuario decidirá que modo desea utilizar, dependiendo cual seleccione se mostrara una pantalla u otra.

- Modo manual.

  Se muestra un interruptor de 2 posiciones, identificado en color gris cuando se encuentra apagado y en verde cuando se encuentra conectado.

- Modo programado.

  Se muestra una pantalla donde en primer lugar se podrá introducir la hora y minuto del día en que se pretende activar el sistema, y en segundo lugar se deberán introducir los minutos y segundos que se desee que esté funcionando el sistema. Cuando se hayan seleccionado estos parámetros, se debe pulsar el botón "Guardar", el cual mandará la información a la base de datos.

5.1. Casos de uso.

Esta aplicación supone el medio mediante el cual el usuario podrá interactuar con el invernadero, es decir, la interfaz hombre-máquina.

La interacción con la aplicación se describe en el siguiente diagrama (Diagrama 5.1.1.) de casos de uso:

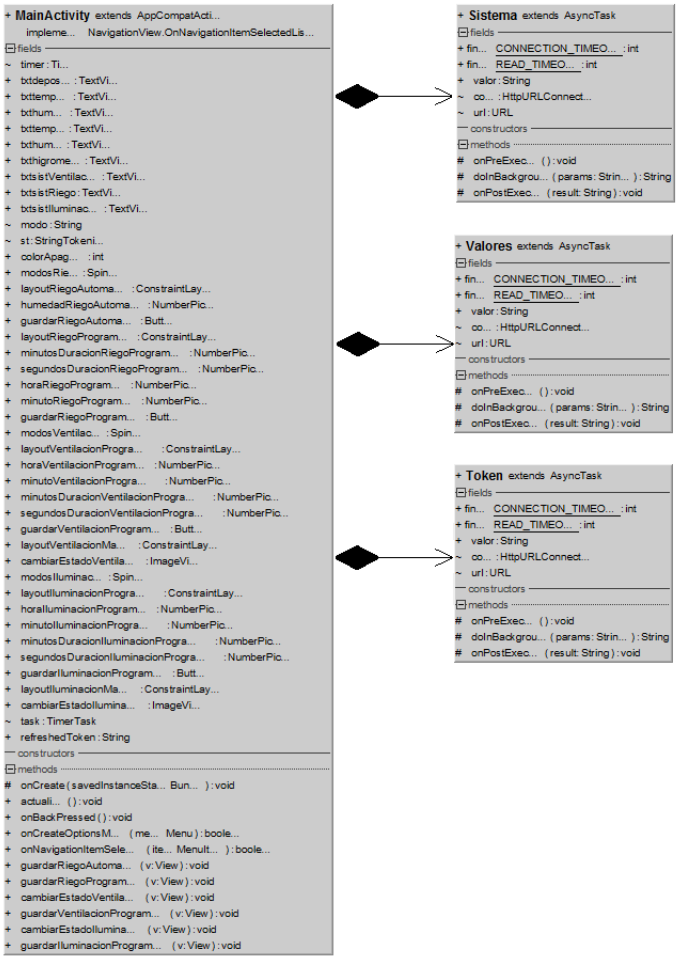
Diagrama 5.1.1. “Casos de uso de la aplicación”.

Fuente: Elaboración propia.

5.2. Clases.

Para la programación de esta aplicación se han utilizado 4 clases, las cuales se representan en el siguiente diagrama (Diagrama 5.2.1.) de clases:

Diagrama 5.2.1. "Clases de programación usadas en la aplicación".



Fuente: Elaboración propia.

- ***MainActivity.***

  Esta clase es el pilar central de la aplicación, donde se manejan todas las funcionalidades principales de la aplicación.

  El método más destacable de esta clase es "actualizar()". Este método se encarga de solicitar los datos al servidor que se van a visualizar. Para ello, este método se repite cada 2 segundos en un hilo paralelo al funcionamiento principal de la aplicación.

- ***Valores.***

  Esta clase es principalmente utilizada por el método "actualizar()". Esta clase hace una llamada al archivo "valores.php" alojado en el servidor, el cual cuando es llamado, imprime los valores de los sensores, además del estado de los sistemas

de ventilación, riego e iluminación. También es utilizada cuando se visualiza alguna pantalla de los sistemas de riego, iluminación y ventilación para conocer el estado de estos y el modo en el que están operando.

- ***Sistema.***

  El funcionamiento de esta clase es muy similar al de la clase "Sistema". En este caso, en lugar de recibir datos, se envían. Es utilizada por los sistemas de ventilación, riego e iluminación.

  Para la explicación de su funcionamiento se expone un ejemplo. Cuando entramos al menú del sistema de riego, encontraremos activado el modo automático con un umbral "x" seleccionados, esto significa que el sistema está funcionando en modo automático con ese umbral seleccionado (Estos datos son recogidos por la clase "Valores"). Entonces seleccionemos el modo programado, introduciremos la hora y la duración para el riego y pulsaremos el botón de guardar. Cuando pulsamos el botón, se hace la llamada a esta clase, la cual recoge los datos y los envía al archivo "sistemas.php".

- ***Token.***

  Esta clase tiene un funcionamiento prácticamente idéntico a la clase "Valores".

  Se utiliza para guardar el id del dispositivo que se obtiene para identificar el dispositivo cuando se envían notificaciones.

  Esta clase se encarga de actualizar este id en la base de datos en caso de que cambie.

5.3. Interfaz gráfica.

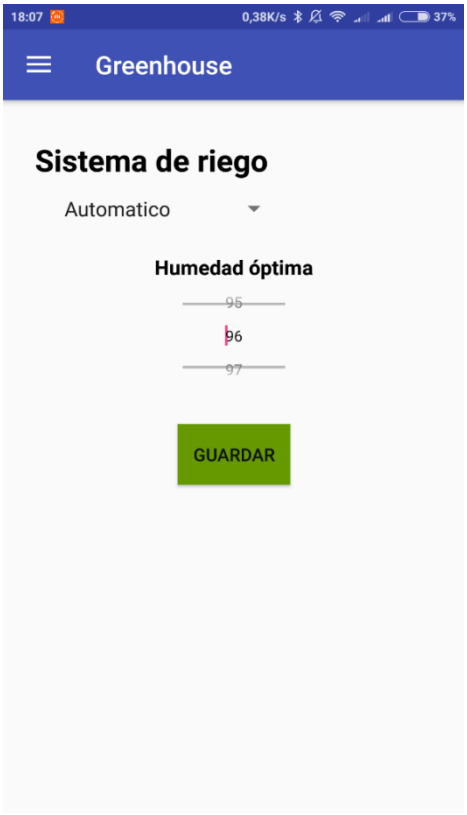
En este punto se pretende mostrar la interfaz gráfica de la aplicación.

- *Datos en tiempo real (Ventana principal)*

- *Cámara de seguridad*

- *Sistema de riego*

  - *Automático*



  - *Programado*

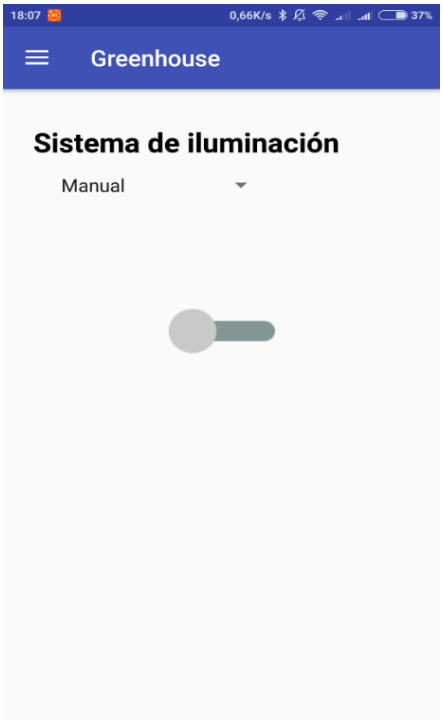
- *Sistema de ventilación*

  - *Manual*

  - *Programado*

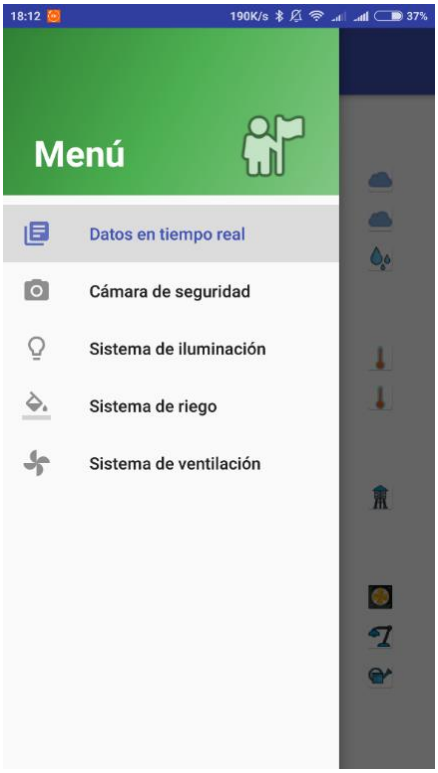
- *Sistema de iluminación*

  - *Manual*



  - *Programado*

- *Menú deslizante*



5.4. Permisos.

Para el desarrollo de aplicaciones en dispositivos “Android” existe un archivo llamado “AndroidManifest.xml”.

Este archivo recoge todos los permisos que se le otorgarán a la aplicación que programemos, de forma que si se quiere utilizar un recurso y se programa la aplicación para ellos, no podrá utilizarse a no ser que se establezca el permiso para dicho recurso en este archivo.

En el desarrollo de esta aplicación se han proporcionado los siguientes permisos:

- Acceso a internet

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

- Notificaciones

```
<service
    android:name=".MyFirebaseMessagingService">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>

<service
    android:name=".MyFirebaseInstanceIDService">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
    </intent-filter>
</service>
```

6. CONCLUSIONES.

6.1. Resultados.

El sistema ha sido puesto en marcha para su prueba, la cual se puede decir que ha sido bastante exitosa.

En la siguiente imagen se puede ver el producto final de la maqueta del proyecto:

El sistema continúa funcionando 24h al día sin problemas, además de surtir efecto en el crecimiento de las plantas.
Se comenzó el proyecto plantando 3 plantas de pimientos habaneros de 6-7cm de altura, las cuales habían sido sembradas en un semillero al exterior, lo que les llevo alrededor de 13 semanas crecer hasta ese tamaño.
Al cabo de 4 semanas dentro de la estructura, se ha observado un notable crecimiento, ya que han crecido hasta los 19-20cm de altura.

No se puede comprobar la eficacia real del proyecto en el crecimiento de la plantas, ya que no se ha realizado ningún cultivo en exterior para poder compararlos, pero como ya se ha comentado, es visible un crecimiento bastante pronunciado desde que se trasplantaron al interior, que bien puede verse influido también por el ciclo de crecimiento de la planta, el cual puede que no sea uniforme.

En los datos observados de las mediciones del proyecto, se ha podido observar que normalmente la temperatura interior con respecto a la exterior suele variar entre 1 y 2ºC, también la humedad, la cual en el interior suele ser alrededor de un 25% superior en el interior de la estructura.

6.2. Trabajo futuro.

Como se comentó anteriormente, este proyecto se ha visto limitado por presupuesto y tiempo disponible, siendo originalmente un proyecto mucho más ambicioso el cual incluso podría haberse visto enfocado a una maqueta a pequeña escala de un sistema de control de invernaderos para grandes producciones. Por lo que en los requisitos futuros para ampliar o mejorar el sistema se incluyen parte de la funcionalidad original de este proyecto:

- Representar gráficamente variables en tiempo real.
- Sistema de multilenguaje dinámico, que permita añadir idiomas a la aplicación sin necesidad de actualizarla.
- Mejorar la estética de la aplicación.
- Incluir un sistema que permita medir con cierta aproximación las horas de luz solar recibida

- Incluir un sistema que permita aumentar la temperatura del interior de la estructura.
- Página web para el control del sistema.
- Conexión a internet de la instalación inalámbrica (actualmente cableada).
- Incluir un diseño modular de instalaciones o sectores que permita controlar los entornos en diferentes áreas con una misma aplicación.
- Adición de perfiles personalizados para distintos tipos plantación.

7. BIBLIOGRAFÍA.

[1] Definición de Arduino. Disponible en "*Arduino".* < https://www.arduino.cc/ > Consultado el 06/04/2017.

[2] Definición de GitHub. Disponible en *"EtherCard".* < https://github.com/jcw/ethercard> Consultado el 10/05/2017.

[3] Definición de Higrómetro. Disponible en "*Wikipedia".* <*https://es.wikipedia.org/wiki/Higr%C3%B3metro* > Consultado el 10/04/2017.

[4] Definición de IOT. Disponible en "*Wikipedia".*
< https://es.wikipedia.org/wiki/Internet_de_las_cosas > Consultado el 02/04/2017.

[5] Definición de MySQL. Disponible en "*MySQL".* <*https://www.mysql.com/* > *Consultado el 16/04/2017.*

[6] Definición de Rapsberry Pi 2B. Disponible en "Rapsberry Pi Foundation". < https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ > Consultado el 09/05/2017.

[7] Definición de Script. Disponible en "*Wikipedia".* <https://es.wikipedia.org/wiki/Script > Consultado el 15/04/2017.

[8] Definición de Servidor Web. Disponible en "*Wikipedia".* <*https://es.wikipedia.org/wiki/Servidor_web* > Consultado el 22/04/2017.

[9] Documentación de FCM. Disponible en *"Firebase Cloud Messaging".* <https://firebase.google.com/docs/cloud-messaging/?hl=es-419> Consultado el 11/05/2017.

[10] "INTERNET DE LAS COSAS [PARTE 2] – SUBIR LOS DATOS A UNA BASE DE DATOS" en geekytheory.

<https://geekytheory.com/internet-de-las-cosas-parte-2-subir-los-datos-a-una-base-de-datos > Consultado el 28/02/2017.

[11] Méndez Clará, Néstor O. (2015). *"Invernaderos automatizados para el desarrollo de la agricultura familiar en el Marco de la Seguridad Alimentaria"* en *Escuela Especializada en Ingeniería ITCA-FEPADE,* vol. 6, nº 6, p. 1-6. <http://www.redicces.org.sv/jspui/bitstream/10972/1859/1/1.pdf> Consultado el 22/04/2017.

[12] Tapia M. Andrea. (2014). *"El potencial de los invernaderos inteligentes para hacer más eficiente el uso de recursos".* Santiago de Chile: El Mercurio. <http://www.elmercurio.com/Campo/Noticias/Noticias/2014/01/29/Invernaderos-inteligentes-una-herramienta-para-hacer-mas-eficiente-el-uso-de-recursos.aspx> Consultado el 22/04/2017.

[13] "TUTORIAL RASPBERRY PI – 7. ESCRITORIO REMOTO VNC + NO-IP" en geekytheory. <https://geekytheory.com/tutorial-raspberry-pi-7-escritorio-remoto-vnc-no-ip/> Consultado el 23/02/2017.

[14] "TUTORIAL RASPBERRY PI – 15. INSTALACIÓN DE APACHE + MYSQL + PHP" en geekytheory. <https://geekytheory.com/tutorial-raspberry-pi-15-instalacion-de-apache-mysql-php> Consultado el 27/02/2017.

[15] Valera Marínez, Diego Luis y Molina Aiz, Fº Domingo. (2008). *"Evolución tecnológica de los invernaderos".* España: Editorial Phytoma. <http://www.phytoma.com/tienda/articulos-editorial/234-199-mayo-2008/3815-evolucion-tecnologica-de-los-invernaderos> Consultado el 22/04/2017.

[16] "Video STREAMING LIVE! CON RASPBERRY PI Y PLAYSTATION EYE TOY" en geekytheory. <https://geekytheory.com/video-streaming-live-con-raspberrypi-y-playstation-eye> Consultado el 15/03/2017.

UNIVERSIDAD DE ALMERÍA

Resumen

El principal objetivo de este TFG es el de la creación de una maqueta de un invernadero que nos permita automatizar ciertas tareas, además de poder visualizar y controlar las variables del entorno.

Para su funcionalidad se ha desarrollado un sistema IOT el cual tiene como núcleo una Raspberry Pi actuando como servidor, y para su control y monitorización se ha desarrollado una aplicación para dispositivos Android.

En el proyecto se han explicado los distintos componentes, tanto “hardware” como “software” que han sido necesarios para la realización del mismo, además de los mapas de conexiones y diagramas de circuitos diseñados para este sistema.

Abstract

The main objective of this TFG is the creation of a model of a greenhouse that allows us to automate certain tasks, besides being able to visualize and control the variables of the environment.

For its functionality I have developed an IOT system which has as core a Raspberry Pi acting as a server, and for its control and monitoring I have developed an application for Android devices.

In this project are explained the various components, both "hardware" and "software" that have been necessary for its realization, as well as the connection maps and circuit diagrams designed for this system.

GRADO EN INGENIERÍA INFORMÁTICA, 2016/2017